Amendments to the Claims:

This listing of claims replaces all prior versions, and listings, of claims in this application.

Listing of Claims:

1. **(Original)** A method for reducing the number of exploitable vulnerabilities in a software application, said method comprising:

creating a vulnerability knowledge database comprising one or more classes of known software vulnerabilities;

applying a code parser to the software application to generate an abstract syntax tree;

comparing the abstract syntax tree and the classes of known software vulnerabilities to identify a set of potential exploitable software vulnerabilities; and

performing a static analysis of the set of potential exploitable software vulnerabilities wherein the static analysis is flow sensitive analysis of a list of constraints, and wherein the results of the static analysis comprise a set of exploitable software vulnerabilities.

2. **(New)** The method of claim 1, further comprising:

performing a dynamic analysis of the set of exploitable software vulnerabilities to identify one or more false positives in the set of exploitable software vulnerabilities; and

discarding the one or more false positives from the set of exploitable software vulnerabilities.

3. **(New)** The method of claim 2, wherein performing the dynamic analysis comprises executing the set of potential exploitable software vulnerabilities with a maximal number of testing configurations.

4. **(New)** The method of claim 1, wherein the vulnerability knowledge database is expandable.

5. **(New)** The method of claim 1, wherein the set of exploitable software vulnerabilities comprises one or more of a security vulnerability, a safety vulnerability, or a reliability vulnerability.

6. **(New)** A system for reducing the number of exploitable vulnerabilities in a software application, the system comprising:

a vulnerability knowledge database comprising one or more classes of known software vulnerabilities;

a code parser that generates an abstract syntax tree from the software application;

a vulnerability code analyzer that compares the abstract syntax tree the classes of known software vulnerabilities to identify a set of potential exploitable software vulnerabilities; and

a static analysis tool that performs a static analysis of the set of potential exploitable software vulnerabilities wherein the static analysis is flow sensitive analysis of a list of constraints, and wherein the results of the static analysis comprise a set of exploitable software vulnerabilities.

7. **(New)** The system of claim 6, further comprising a dynamic analysis tool that performs a dynamic analysis of the set of exploitable software vulnerabilities to identify one or more false positives in the set of exploitable software vulnerabilities, wherein the one or more false positives are discarded from the set of exploitable software vulnerabilities.

8. **(New)** The system of claim 7, wherein the dynamic analysis tool executes the set of potential exploitable software vulnerabilities with a maximal number of testing configurations.

9. **(New)** The system of claim 6, wherein the vulnerability knowledge database is expandable.

10. **(New)** The system of claim 9, further comprising a user interface that enables a user to enter an additional known software vulnerability to the vulnerability knowledge database.

11. **(New)** The system of claim 6, wherein the set of exploitable software vulnerabilities comprises one or more of a security vulnerability, a safety vulnerability, or a reliability vulnerability.

12. **(New)** A system for reducing the number of exploitable vulnerabilities in a software application, the system comprising:

a vulnerability knowledge database comprising one or more classes of known software vulnerabilities;

a code parser that generates an abstract syntax tree from the software application;

a vulnerability code analyzer that compares the abstract syntax tree the classes of known software vulnerabilities to identify a set of potential exploitable software vulnerabilities;

a user interface that presents the set of potential exploitable software vulnerabilities to a user and enables the user to select one or more potential exploitable software vulnerabilities from the set of potential exploitable software vulnerabilities; and

a static analysis tool that performs a static analysis of the selected ones of the set of potential exploitable software vulnerabilities wherein the static analysis is flow sensitive analysis of a list of constraints, and wherein the results of the static analysis comprise a set of exploitable software vulnerabilities.

13. **(New)** The system of claim 12, further comprising a dynamic analysis tool, wherein the user interface presents the set of exploitable software vulnerabilities and enables the selection of one or more of the exploitable software vulnerabilities from the set of exploitable software vulnerabilities, and wherein the dynamic analysis tool performs a dynamic analysis of the selected ones of the set of exploitable software vulnerabilities to identify one or more false positives in the set of exploitable software vulnerabilities, wherein the one or more false positives are discarded from the set of exploitable software vulnerabilities.

14. **(New)** The system of claim 13, wherein the dynamic analysis tool executes the selected ones of the set of potential exploitable software vulnerabilities with a maximal number of testing configurations.

15. **(New)** The system of claim 12, further comprising a dynamic analysis tool that performs a dynamic analysis of the set of exploitable software vulnerabilities to identify one or more false positives in the set of exploitable software vulnerabilities, wherein the one or more false positives are discarded from the set of exploitable software vulnerabilities.

16. **(New)** The system of claim 15, wherein the dynamic analysis tool executes the set of potential exploitable software vulnerabilities with a maximal number of testing configurations.

17. **(New)** The system of claim 12, wherein the vulnerability knowledge database is expandable.

18. **(New)** The system of claim 17, wherein the user interface enables the user to enter an additional known software vulnerability to the vulnerability knowledge database.

19. **(New)** The system of claim 12, wherein the set of exploitable software vulnerabilities comprises one or more of a security vulnerability, a safety vulnerability, or a reliability vulnerability.